

Archive

The subscription magazine for *Archimedes* users

Volume 1 • № 1 • October 1987

Price £1.20

Contents

News & Comment	2
Hardware & Software Summary	3
Hints & Tips	5
Disc File Transfer	7
Diary of an Archimedes User	9
BBC Micro RS423 Link	10
ROM-Link Packages	20
Clares Micro's ToolKit Review	22
Archimedes – The Music Computer	26
Letters Page	27
Structured Programming with BASIC V	28
Technical Notes on BASIC V	32
Graphics Demonstration Programs	38
Screen Mode Selection Chart	42
User Guide Errata	42
Fact-File	45

*HELP Archive

What are its aims? – Archive is a subscription magazine, for users of the Acorn Archimedes range of computers, which aims (1) to provide information to the user (2) to provide a forum in which we can all share ideas (3) to give the benefit of bulk buying of software (4) to allow software and hardware vendors to advertise their wares.

Is it a User Group? – No, but I would like it to have a "User Group feel" – like BEEBUG when it first started – except that Norwich Computer Services has to earn a living from the magazine; hence the "Goodies List" on the back page.

However, Sue and I are not in this business to make lots of money; we enjoy the work we do and it's very satisfying to be able to provide a useful service. We only ask to make enough money to live on plus a bit to give away and then we'll be happy. (I hope it doesn't sound too trite, because it's true.)

HELP! – Could you help us, please, to make Archive a success? We don't have the muscle or the financial budget of the big magazines and cannot afford to do a vast amount of advertising, so, if you think Archive is good, please take out a full subscription (if you have not already done so) and recommend the magazine to a friend or two. If you want any more information sheets and subscription forms, please let us know.

Can we answer technical enquiries by phone? As much as we would like to continue the policy we have always had of being available to answer all your technical enquiries by phone, we felt that if we were not careful, we would be so inundated with calls that we would not have time to get the information out to you through the magazine. For that reason, we have introduced the Technical Help Service so that

those who really need the instant access to help can purchase it. (£8 per annum.) I hope you will bear with us and, if you do not feel it is worth the extra £8 to you, send your enquiries by post.

Do we take Access or Visa? No, I'm afraid not – mainly because of the high percentage that we would have to pay for the privilege (I think it's 6% when you first start). The other reason is that we have always had a policy of sending goods out by return of post, whenever possible, regardless of whether the cheques have cleared through the bank. The way we see it is that if you trust us by sending a cheque without the Access guarantee, it is reasonable for us to trust you by sending out the goods without waiting for the cheque to clear. Perhaps we are foolish to take the risk, but we have only had two dud cheques in three years of full time trading.

Although there's only two of us here, we are surrounded by a lot of people who help us in various ways, some voluntarily and some professionally, without whom we would not be able to stay in business. I don't think it is appropriate to mention them all by name, but I would just like to assure them that we are really grateful to them all. However, as I have said in most of my previous publications, as a committed Christian, there is One Person who never fails us and without whom we would achieve nothing worthwhile. God supplies all our needs, and we thank and praise Him!

I hope you find Archive interesting and informative and that you will feel able to contribute your own ideas, hints and tips or even full articles. I'm afraid that we cannot afford to pay the vast sums for articles, but at least you would automatically become an "HLMTHS". – Honorary Life Member of the Technical Help Service!

Archive

Volume 1 • № 1 • October 1987

What's in this issue?

We've tried to give a wide range of articles in this first issue. Obviously, since the machine has only just been launched, we spend a lot of time discussing the teething problems and ways round them. Then there are articles looking at some of the new features, in particular two articles about BASIC V, one which is highly technical and the other aimed at less experienced programmers who wants to take advantage of the new structuring facilities which BASIC V provides.

We have a couple of reviews, although at this stage there is not a lot available to be reviewed! We have also allowed a couple of companies to tell us in a bit of detail about the projects they are working on for the Archimedes.

What about succeeding issues?

There will be lots more reviews and more detailed and careful explanation of the new facilities of the Archimedes and also the new concepts – e.g. the "WIMP" environment (Windows, Icons, Mouse & Pointer), the Arthur supervisor, relocatable modules, etc, etc.

Let us know which bits you find difficult and we'll try to explain them. Let us know too if you find anything particularly interesting. Send in your programs and your hints and tips and we'll see if we can include them.

Share-Ware Software Scheme

How about us implementing this American idea called "Share-Ware"? You send in the programs you have written, we set up discs with several programs on each and sell them through the magazine, the price we charge being the cost of the blank disc plus something for our time and effort in copying and distribution. Then each program has, if the author feels so inclined, a few REM statements at the beginning giving their name and address and a suggested contribution to send to them if you think it's a useful program. That way they can also send you any modifications they make to the program or any extra documentation available.

Surely this has to be a sensible way of avoiding too many people hacking away in their own back rooms (or offices or laboratories) writing programs for exactly the same applications. If you can start off from a program someone else has written (hence the importance of writing well-structured programs), you can perhaps improve on it and pass the message back down the line. I doubt that this idea will prove too popular with the software vendors, but I haven't definitely decided to set up this Share-Ware system. Let me know what you think. **A**

Comment – Where has all the software gone?

It does not take much knowledge or intelligence to work out that the Archimedes is a tremendously powerful computer and that it has amazing potential, but where is the software to run on it? Quite a few companies say that they have "Archimedes compatible" software – in other words it is written for the Master or Master Compact and it happens to work on the Archimedes too.

But where is the software that will utilise some of this raw processing power that is sitting on my desk, as yet unused? There are a number of companies who seem to be investing huge amount of development time and money producing these packages, but as yet there is not a lot to show. Could it be that it is taking them longer than they, or Acorn, thought to get to grips with the RISC technology, or could it be that they are having problems with things like operating system bugs and ADFS bugs which inevitably start to crawl out of the woodwork as soon as you start to develop worthwhile programs on a new computer system.

I guess we'll just have to be patient. I have every confidence that the Archimedes is as big a leap forward as the original BBC micro was back in 1981. I for one look forward to being heavily involved with this exciting new machine and wish Acorn and all the other companies involved, all the very best with this new project. **A**

News, News, News.....

News from Acorn

The 0.3 Operating System and Welcome Disc "was due a few days ago" (mid-September) and is now "due any day", but I can't find out what the policy is for up-grading. In theory, having filled in your registration document (you have filled it in and sent it off, haven't you?!) you should get the up-grade automatically. Those of you who are registered on our Technical Help Service can contact us to find out the latest ETA on the 0.3 Arthur. (Please remember to quote your THS number, not your subscription number, when making an enquiry. Thanks.)

The Programmers' Reference Manual is not available yet. The official Acorn line seems to be that it won't be in print until November but as soon as it is, we hope that we might be able to offer it for sale through the magazine. In the meantime, we have a copy of the draft version of it, so if THS folk have specific enquiries, give us a ring, remembering to quote your THS number.

Software Received

Archimedes Tool-Kit Module from Clares Micro Supplies. This is a pre-release version 3.50 and is reviewed in this issue.

Delta-base from Minerva - To be reviewed in next month's Archive magazine.

Inter-Word, Inter-Sheet, Inter-Chart and Wordwise Plus – Computer Concepts are "having discussions with Acorn" over ways of presenting their software for sale. If they sell it on disc, it is so easily copyable that they fear considerable loss of income as a result. They would like to sell it in ROM format, but we will have to wait and see.

Software and Hardware for the Archimedes

All the information about names, addresses and telephone numbers are in the *Fact-File* at the back of the magazine.

AcornSoft Ltd – I have so far been unable to get any further information than is given in the Archimedes Software Applications Catalogue dated June 1987, i.e. that they are doing the View family under the emulator and View Professional (due November?), Logistix (September?), Comal under the emulator and ANSI C, ISO-Pascal, Fortran-77 Prolog and Lisp plus various terminal emulators, as well as, of course, Zarch, which we are all waiting for with bated breath. Hopefully, much of this software will be launched at the PCW Show which is presumably why I can't get hold of anyone this week (September 19th).

Audio-Visual Publication, who have a huge list of educational software for the BBC micro – at least a couple of hundred titles – have given us a list of 73 programs which they say are "available for the Archimedes". I have asked to see some of them for review purposes.

BEEBUG Ltd are producing a modem podule and are "hoping to do C" for the Archimedes as well as retailing the machine itself and providing a quite generous trade-in on your old Beeb's, Masters or Compacts. Also, they're thinking about doing *a magazine specifically for the Archimedes!* – I wish them every success!

CJE Micros are offering a range of hardware products covering all the cables you might want, conversion boxes to allow you to use your existing 1451 and 1431 Microvitec monitors, 305 to 310 RAM upgrades, as well as two

different multi-sync monitors (if you've got £600 or so to play with). On the software side, they are translating their well-known and much-loved Font-Aid over to Archimedes.

Clares Micro Supplies are very heavily involved with the Archimedes computer and have already released a pre-release (sic) version of their ToolKit module. (Reviewed in this issue.) They are working on a sophisticated art package called Artisan £39.95 (£37 through Archive) and a database (Alpha Base) should be available soon.

Computer Concepts' DeskTop Publishing package, as yet un-named, sounds more impressive than the DTP system I'm using to prepare this magazine. It operates entirely within the WIMP environment, fully formatted on-screen, scaled full-page view visible while editing, has user defined styles, e.g. Times 14, bold, underlined etc, then if the definition is changed, all occurrences of that styles change too, multiple windows can be opened on the *same* document, graphics can be placed anywhere, headers, footers and footnotes (all of which can be multi-line, multi-column and can contain graphics!), automatic generation of contents page and index (authors of the Archimedes User Guide, please note!), in-built spelling checker (greatly enhanced version of Spell-Master) containing 60,000 words, checking at 50,000 words per minute, intelligent 'guess' facilities, user dictionary, optional thesaurus, supplied on ROM! Wow! I can't wait, but I'm going to have to as it scheduled for "Early 1988"!

Computer Concepts' object-orientated drawing package again sounds more impressive than the stuff I use on the Apple Macintosh. Works within the WIMP environment, wide variety of pre-defined object shapes available,

filled or un-filled, with or without lines around them, objects can be grouped and treated as a single object, any object can be re-sized, rotated, change colour fill etc, each drawing can be any number of layers, drawings can be scaled and worked on at the new scale, drawings can be pasted into the word-processor, an optional language compiler will allow you to 'draw by program' then modify by hand, e.g. you can draw a statistical graph from supplied data then modified by hand. Again this program is scheduled for "Early 1988".

They seem to have dropped plans for an all-singing, all-dancing Disc-Doctor II but are trying to arrange to make Inter-Word, Inter-Sheet and Inter-Chart available as well as Wordwise Plus to fill in the gap.

ESM Software for Schools have a version of their Supastore database aimed at school administration, pupil records, library management etc.

Intelligent Interfaces are the first (as far as we know) to produce an interfacing podule for the Archimedes. However, this is not for the 'interested amateur'. This is a professional engineering standard IEEE488 capable of communicating with fourteen IEEE488 test instruments. It can transfer data at up to 250kbytes/second and comes with high level software support. It can be configured either as a system controller or as a simple talker/listener. The price is £326.50 incVAT.

Meadow Computers have converted their Micro-Trader Integrated Accounts System for use with the Archimedes. At £230, it seems good value if it comes up to the impressive specification. It also links in with their Micro-Stock and Micro-Mailer programs and also with Micro-Aid's Extended Payroll. The BBC micro version has received some quite favourable

reviews on and, at the price, it has to be worth considering. Provided it is flexible enough for your business, you could have quite a good accounting system without it costing you an arm and a leg.

Minerva Systems (See advert for details) DeltaBase (£29.95) is available now. We hope to review it next month. Minotaur (£14.95) is available now and will be reviewed next month. System Delta-Plus (£69.95) which sounds like an extremely impressive WIMP driven database package is due in October. Accounts modules (£69.95 per module) – all five modules should be available some time in October. System GammaPlot, GammaPlus and System Sigma are all scheduled for November.

Tube-Link – For those of you who haven't actually got hold of an Archimedes yet, how about getting Tubelink's "Advanced BASIC" which is an extension to Hi-BASIC for the 6502 second processor on the BBC micro. You can then write programs ready to transfer to your Archimedes when it arrives. This extension which was apparently originally called "Archie BASIC" (I wonder why they changed it?!) is said to be "compatible with Acorn's new Archimedes BASIC" – it certainly seems to have most of the new commands, operators, trace and error handling routines that were introduced in BASIC V. (£29.95 on disc or £34.95 on EPROM.) **A**

Hints and Tips

Using Electrohome Monitors

Ian Nicholls of Kidderminster says that the "Electrohome" colour monitor, as sold by Opus, works OK on the Archimedes. All you have to do is make up the appropriate connector. (Or purchase one from CJE Micros or somesuch.) The pin connections for the Archimedes are given on the inside back cover.

Using Modes 18 - 20

Whilst on the subject of monitors, let me clear up the confusion there seems to be in the computer press about the high resolution modes and the extra video hardware which the 400 series machines are said to have. This extra hardware is NOT for driving modes 18 - 20. Even on the 300 series, there is all the hardware needed to run these modes on a multi-sync monitor.

The extra hardware in the 400 series is actually foreseen higher resolution graphics—it will go up to 1280 by 976 pixels in monochrome (160 characters by 122 lines of text) and, as with modes 18 - 20, it requires a special monitor. There is no industry standard for such monitors, and Acorn are still looking into providing a suitable monitor with their own badge on it. This extra high resolution is of course aimed at the CAD (computer aided design) market, and it is said that AutoCAD is being translated onto the Archimedes.

Using a Multi-sync Monitor

If you have a multi-sync monitor and want to get it working on the 300 series, all you need to do is to switch the computer on whilst holding down the **<R>** key. This resets the CMOS RAM settings (so if, like me, you had changed them, you will have to re-configure again!) and it changes the configuration of the RGB output to

run in multi-sync mode. It then allows you to access the extra screen modes that had been previously forbidden to those of us who only have standard monitors. If you should want to change the machine back again to run a standard monitor, you have to do another **<R-power-up>** which switches it back again (and re-sets the CMOS RAM settings AGAIN!). In other words each time you do an **<R-power-up>** it toggles between the two monitor settings.

Using View

Gerald Jones of Northampton tells us that only those of you who have gone from a Master Compact to an Archimedes will be able to get View going—for the time being, at least, because the ROM versions of VIEW will not work under the emulator. If you do have a disc version of View, Acorn tell us that the way to get it going is to copy 65ARTHUR from the MODULES directory on the Archimedes Welcome disc onto another disc and VIEW from the Compact Welcome disc and then use *BUILD !BOOT to create a boot file which consists of:

```
*65ARTHUR
*GO F800
*LOAD VIEW 8000
*GO 8000
```

and then remember to type *OPT 4, 3 to set it up so that it will execute the boot file on **<shift-break>**. All you have to do then is to chop up your VIEW keystrip and stick it together in the Archimedes keyboard format remembering that the f0 key functions are now on the **<print>** key!

Attaching a 5.25" drive

Have you tried to get a 5.25" disc drive linked up to the Archimedes? Presumably, you did the obvious thing and removed the cable from the p.c.b. to the 3.5" drive and plugged in the cable from the 5.25" drive. And you too found that it didn't work! Well, the official answer is that it is

possible to connect a 5.25" drive but that there are "different connections". Actually, it seems that the only difference is that they have put the connector in the opposite way round!

Unfortunately, you can't just turn the connector round because of the location lug on the side of it. The solution is either to file the notch off the side of the connector so that it will go in either way round, or buy a new 34-way insulation displacement connector which you can crimp onto the end of your disc cable, the opposite way round from the existing one, or buy a disc extension cable from a supplier such as CJE Micros (£15 + £1 p&p – address in Fact-File at the back of the magazine). The advantage of doing it that way is that the extension cable is long enough to stick out at the back of the computer through the podule connection slot so that you can connect and disconnect the extra drive without taking the lid on and off.

You can, if you want, put a single 5.25" drive as drive 1 or a pair of drives as drives 1 and 2. The extra drives must, of course, have their own power supply units and you need to type *CONFIGURE FLOPPIES 2 (or 3) so that the ADFS knows how many drives there are. You will also need to get inside the 5.25" drives in order to change the link settings to configure them as drives 1 (and 2) and you should also, theoretically, remove the terminating resistors – usually a set of resistors (dual-in-line or single-in-line) near the drive-select setting links. This is because there are already terminating resistors in the internal 3.5" drive and you should not have two sets. If you have a dual drive where they are on top of one another rather than side by side, you may find it difficult to get at the second drive in order to change the link selection. If so, the easiest solution is to make the top drive 2 and just remember that the lower drive is drive 1.

One problem you may still find (which Acorn haven't yet managed to explain) is that if you switch on the power to the disc drive before or at the same time as the computer, the computer seems unable to start up – you just get a blank screen. All you have to do though is to make sure you switch the computer on first and then the disc drive. You will also find that the desk-top program does not recognise the third drive – even if you have typed *CONFIGURE FLOPPIES 3, it only displays two disc icons on the screen. (If anyone works out how to modify the program to recognise all three, let me know.)

ADFS bugs

There are one or two bugs in the ADFS, though they are apparently reasonably obscure. The only one I have found myself is that if you try to *BUILD a !BOOT file onto a disc that already has a !BOOT file and you have left the disc write-protected, when you press <escape>, the system hangs up and you have to press <ctrl-break> or reset to escape. Some of the bugs have been corrected on the 0.03 version of the ADFS that you will find in the MODULES directory on your Welcome disc. This version will have to be *RMLOAD'ed, then to check that you have got the right version installed, type *HELP MODULES which gives you the current version numbers of all modules. It should say ADFS 0.03 (17 Jun 1987) instead of ADFS 0.02 (05 Jun 1987). Incidentally, the file name on the Welcome disc is Adfs0-03, not Adfs0_03 as you might tend to expect, so if you get a file not found error, check that you have typed it correctly.

Have you discovered...?

I'd like this to be a regular feature. It consists of the sort of things that may be obvious to some people, but perhaps that you've noticed from reading the User Guide (or the Programmers' Reference Manual when it becomes available). If you find something that you thought, "Well, I didn't realise that!", let us know. I'll start it off this month with things that were new to me.

- The pad character "l" which is the shifted back-slash character (the key just above the return key) can be used to "pad" out those incredibly long VDU codes that end with lots of zeros. For example, to switch off the cursor, you can use VDU23,1,0,0,0,0,0,0,0,0 which can admittedly be shortened to VDU23,1,0;0;0;0; can now be reduced to VDU23,1l which has the extra advantage that you don't have to try to remember exactly how many zeros it needs to complete the command.
- Instead of saying value=value+extra, you can use the new operator "+=" and say value+=extra which is a significant saving if, like me, you tend to use long variable names; likewise with -=extra. (Well with "604412 bytes remaining", it may not be worth worrying about memory space, but it does save on printer ribbons!)

• DANGER! • BEWARE! • WARNING!!!!

Newcomers to ADFS beware! If you have been using *WIPE * and looking through and saying "yes" or "no" to each, **Don't try it on ADFS!** You will find that *WIPE * deletes ALL the unlocked files in the current directory. It's like *DESTROY * but without even displaying a list of all the files it was about to delete and asking for confirmation. I discovered this 'nasty' the hard way – I lost about 10 programs at a stroke. It wasn't too bad though as I had copied them across from DFS and only modified one or two.

What you should say is *WIPE * C where the C stands for "Confirm" – then it asks you about each file in turn. – **You have been warned!!!!**

Disc File Transfer

If you have programs and files on other Acorn machines that you want to transfer across to the Archimedes, what is involved? Well, it depends what format the data is currently in.

- **Master Compact** owners are the best off because your discs are readable on the Archimedes which can operate directly on either 640k ADFS discs or its own new standard 800k ADFS format. (You also have a word-processor you can use – View, on the Welcome disc, works under the 6502 emulator – see Hints and Tips.)
- **Master** users with 5.25" ADFS discs will again be able to get the Archimedes to read their discs directly if they can get their disc drive linked up to the Archimedes. (See Hints and Tips.) If your files are on DFS on the Master, you should find a program on the Master Welcome Disc called DIRCOPY which allows you to transfer files from DFS to ADFS prior to transferring them to the Archimedes.
- Owners of straight **B's and B+'s** using DFS have more of a problem. If you can make up an RS423 cable, or buy one from CJE Micros, you can use Dr Alpiar's programs to transfer files one at a time. This is rather time consuming because unless you happen to have an Archimedes on which the RS423 control lines work properly you will have to run at low baud rates. You could however modify the program to take a whole set of file names and send the files one by one while you go and have a cuppa!
- The easy way is to let us do it for you! Send us copies of the discs you want transferring (Please don't send originals!) and we will put the files onto 3.5" ADFS for £5 per disc, inclusive. 

Build your own system...

with the Minerva SYSTEMS DELTA, GAMMA and SIGMA

Individual programs designed to integrate, use separately or link together to form a powerful business system. Systems Delta, Gamma and Sigma... designed by highly acclaimed professional programmers with proven ability on the BBC, now providing definitive software for the Archimedes

ARCHIMEDES

DeltaBase £29.95

Extremely powerful yet simple to use general Database. Over 8000 records, 255 fields and 16 subsets. Extensive single or multiple criteria search, sort and mass facilities. Restructuring records is easy. Imports data from Viewstore, Betabase, Interbase, Mini-Office, etc.

System DeltaPlus £69.95

Extensive Relational Database Management System. 2 billion records, 16 million fields, 26 million subsets. Infinite files open at once. Powerful WIMP database. Multiple user definable print formats. Univalued, comprehensive maths, searches and sort. Mailing labels. Extensive Reporter. Card copier and sort. Fully programmable.

GammaPlot £34.95

Versatile Graphics program. Multiple charts on screen at any time including pie, scatter, line, hist, 3D. On Screen editor for text presentation. Macros. Entry by Keyboard or any Delta Gamma file.

System GammaPlus £69.95

Graphics Management System. Multiple charts on screen at any time. Over 10 types available. Definable scaling and screen formats. Screen editor. Statistical analysis. Three dimensional charting. Macros. Fully programmable.

BBC MASTERS

System Delta with Card Index £64.95

Still the most comprehensive and flexible data management system on the BBC. 8160 records, 255 fields, 2000 characters per record. 8 subjects. Up to 5 files open at any time. Multiple criteria searches, sort and maths facilities. Multiple print formats. Default card and card copier for transfer between files. Fully programmable relational system with over 150 commands available.

Mailshot, Reporter and InterView Link £19.95 each

Three versatile support programs for use with all System Delta files. Mailshot produces up to 4 across labels and links with wordprocessors for mail-merged documents and Reporter produces simple lists as well as fully definable reports including negative or positive columns, averages, totals, headings, sub and sub-sub-totals. InterView Link transfers data directly or as a result of any calculation including sum, average, count, etc.

System Delta Personal Accountant £45.95 (Nominal £55.95)

Five comprehensive accounting modules for use separately or as an integrated suite. Many advanced features as listed above. Total flexibility allows customisation if required.

Order Processing/Invoicing, Sales Ledger, Stock Management, Purchase Ledger, Nominal Ledger

Pupil and staff records allow extra fields to be added if desired. Entry reports for examination boards. Form 7 DES statistics, etc.

MICRO

Specialist Applications from £54.95

Comprehensive management programs for Hotelier, Video Rental and Newsagent.

System Gamma £49.95

The first Graphics Management System. Anyone can produce stunning graphics and charts easily from the menu driven GammaPlot supplied. Multiple charts on screen at any one time and in any position desired. On screen text editorial. Several fonts provided for variable text. Statistical analysis. Macro facility. Entry by keyboard or System Delta files. Fully programmable to create your own graphics programs.

COMPUTER SOFTWARE AND SYSTEMS SPECIALISTS

69 SIDWELL STREET EXETER EX4 6PH - TELEPHONE EXETER 0392 37756



Diary of an Archimedes User

Some problems and their solutions.

I'm not trying to criticise Acorn, but I just want to give you a diary of what went wrong with my machine in the first few days and weeks. If you have similar problems you will know that you are not alone! There is someone out here who cares and will try to help you!

The first problem was the **shift keys sticking down**. This was solved very simply by removing all the screws under the keyboard and then loosening the screws holding the printed circuit board into the case. The board could then be pushed slightly upwards until the shift keys were away from the edge of the cut-out and the screws tightened up before screwing the bottom back on. Acorn say that this problem will be sorted out in the production stages.

When I started typing things on the screen, I found that the **text disappeared off the bottom**. So I was tempted to do a *TV0,1 or even 1,1 to pull the screen up again, tut-tutting the fact that Acorn had sent it out configured in a way that didn't suit the monitor that goes with it. But when I ran a few applications and saw that the whole border around the active area of the screen was rather low, I looked at the adjustments available on the monitor itself and found a 'V-position adjust' on the back. A gentle twist on that control and my equilibrium was restored.

Problem three was **how to use EDIT** to edit a BASIC program. "Just type EDIT" said the manual, but with the 0.2 operating system and Welcome disc, the EDIT program needs to be loaded as a relocatable module. To do this, put in the Welcome Disc, type *MOUNT<return> and *DIR MODULES<return> and then *ARMBE<return>. From then on, you can indeed "just type EDIT".

Getting **RAMBASIC** going was the next problem. Having been told that it was so much faster than the ROM version and that it had lots of built-in help information, I wanted to try it out. The solution is to type QUIT<return> (or *GOS<return>) to get into the Arthur operating system, put in the Welcome Disc, type *MOUNT<return> and *DIR MODULES <return> and then *RAM_BASIC<return>. Then if you type HELP<return> (as opposed to *HELP) you will see an explanation of the extra HELP information available.

"No keyboard present" – The one major problem I had was that in the middle of editing a BASIC program one day, the computer froze and would not respond to any key or mouse press including <escape>, <break> <ctrl-break>, or <reset>. Having abandoned any hope of saving the work I had done in editing my program I switched the computer off and on again. It told me "No keyboard present – auto-booting" and proceeded to execute the !BOOT file on the disc! (This illustrates an interesting idea that if you want to run some sort of rolling demonstration program that does not require a keyboard, you can remove the keyboard completely and just switch on and let it boot up the disc in drive 0.)

The solution to this problem was to start to dismantle the computer and get the packing case out and threaten to send it back to Acorn! Actually, having taken out the disc cable, RS423 connector etc etc, I decided that just before I sent it back I would give it one last try. I gave the main printed circuit board the old BBC remedy of "firming the chips into their sockets". That did the trick! It sprang back to life again and is still going strong.

Then it was time to try the **screen dumps** so that we could grace the new magazine with some pretty pictures. Sadly, I could not get any of the screen-dump modules to work. Acorn admitted

that they were rather low on their get-it-working list but said that there would be some working versions on the 0.3 Welcome disc and eventually a single screen-dump program which would work on any of the FX, RX and MX printers.

The final problem, as yet unresolved, relates to the article by Dr Alpiar which follows. I tried to get the RS423 serial port working in order to transfer programs etc from the BBCmicro. Once we had got the pin connections and discovered the need to link together various pins, we managed to get the transfer started. We soon discovered that at 1,200 baud, the transfer went OK but any faster and the data got corrupted.

To cut a long story short, the conclusion that I have come to is that the control lines are not working correctly. If the sending machine is transmitting data faster than the receiving machine can deal with it, a backlog builds up as the buffer gets full and the data stream gets interrupted because the sending machine keeps on sending. Having said that, I know of one person who has successfully got the Archimedes talking to a serial printer which, unless it has a huge buffer, must surely be slower than the machine which is sending. So there seems to be no consistency about what works and what doesn't. Acorn have admitted that there is a problem, that is partly to do with software and partly with faulty chips and that their development team have it as a top priority but, as I write, there is still no solution offered apart from working at very slow speeds.

I've just realised something... If they are only just working on the problem now, while the 0.3 Arthur is being ROM'ed, it won't be correct in that version either! Well, we'll just have to wait and see. Apparently they are preparing an applications note to tell us what to do. The answer will probably be not to try to use the control lines but to use Xon-Xoff protocol! A

Archimedes & BBC Micro in Conversation

Dr Ronald Alpiar

This article is addressed to many (if not most) of you Archimedes owners who already have a BBC or Master. You have probably accumulated a library of BASIC programs, some of which use the BBC Micro Analogue or User ports: and you're positively itching to test them out on your new Archimedes. But there are two obstacles. At present the Archimedes has neither ports, nor means of reading 5.25" floppies on which most of your program are filed.

True, given time, this will sort itself out. An I/O podule will supply BBC Micro type ports: disk drive extensions will appear – or you may even consider buying a 3.5" disk drive for your BBC Micro. But the status of the I/O podule is still "price TBA", and you may feel it somewhat prodigal to buy an otherwise un-needed drive just to bridge a temporary gap.

Luckily an instant (well, almost!) solution to all these problems is at hand. You'll find it at the rear of both machines in the form of connectors labelled 'RS423'. These are the 'serial ports' which are normally used to drive slow serial printers, or for connection to phone lines via a modem.

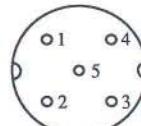
We shall start by identifying the pins on these connectors, explaining what they do, and showing how to connect them so as to allow information to flow between Archimedes and BBC Micro. Next we will turn our attention to the *FX calls which tell the operating systems how to send data across the physical connections we have set up. Then we will look at some BASIC routines designed to test the communications system, pointing out problems

which may occur, and finally, we will give annotated BASIC application programs to effect data file transfer. In the next article, we will show how to use the BBC Micro as an I/O module for the Archimedes.

Plugs and Sockets

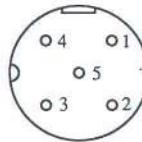
In the accompanying diagrams, we take it that you are facing the solder pins or solder lugs at the rear of a connector, so that the leads from the connector's rear end come towards you. A 'plug' is a connector with protruding pins, which fit into corresponding holes in a 'socket'. (If these pretty obvious points seem to be laboured, it is only because misinterpreting pin-out diagrams is the most common error in electronic construction – and even plagues experienced amateurs – yes, me too! As it happens, it is most unlikely that you'll be able to do any electrical damage by miswiring connections, but it will waste time and be very frustrating if the interconnection does not work first time.

The RS423 connector on the BBC Micro is clearly labelled at the back of the case. The socket is what is referred to as a "DIN 5 pin 360 degree domino (or type C)". So to connect to it what you'll need is a "DIN 5 pin domino (or type C) plug"! Any other DIN plug (e.g. such as is commonly found in cassette connectors) just



Socket

1	RX
2	TX
3	CTS
4	RTS
5	EARTH



Plug

won't fit and the use of force may cause damage.

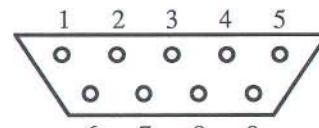
Figure 1 shows the pin connections. Two of the lines are for control signals, two are for data, and earth makes up five. The standard pin function abbreviations in the figure mean:

RX	receive data (input)
TX	transmit data (output)
CTS	clear to send (input)
RTS	ready to send (output)

At the Archimedes end, the RS423 takes the form of a "D-type 9-way PLUG" and not a socket as you might tend to expect. (This is done to avoid mixing it up with the monitor connector which is a 9 pin socket.) As figure 2 shows, the connector includes the functions of the BBC Micro connector, plus four more besides – which are all to do with modem operation:

DCD	data carry detect
DTR	data terminal ready
DSR	data system ready
PLR	phone line ringing

The serial chip assumes that the RS423 is connected to a modem, even though that's not the case here. To fool it into thinking that it's talking to a modem, we shall have to connect the three pins DCD, DTR and DSR together: PLR should remain unconnected.



1	DCD	6	DSR
2	RX	7	RTS
3	TX	8	CTS
4	DTR	9	PLR
5	EARTH		

Figure 1 – View of solder pins on rear of socket and plug on BBC Micro RS423 connectors.

Figure 2 – View of solder pins on rear of socket to fit into Archimedes RS423 connector.

The basic principle for connecting up any two RS423 ports is absurdly simple: The RX of one is connected to the TX of the other and vice versa and the CTS of one is connected to the RTS of the other and vice versa and then earth is connected to earth. That is, we have a simple cross connection of the two data and the two control lines.

To make up the connecting cables you'll need a soldering iron with a fine bit and the following items, which most electronic construction shops should have in stock, or which can be obtained by mail order from such firms as Maplin Electronic Supplies Ltd.:

- 1 off D-type 9 way socket
- 2 off DIN 5 pin domino (type C) plug
- 1 off DIN 5 pin domino (type C) socket
- 1 metre 5-core cable (or 5 differently coloured 1 metre lengths of connecting wire would do.)

In addition, a simple continuity tester, steady hand and eye and calm nerves are a decided advantage.

Using the solder tags at the rear of your D socket, and one of your DIN plugs, connect the 5 pins as indicated, with your 1 metre cable. Remember that the connection diagrams as shown are what you see as you look at the solder lugs. You should also note that the 5 pin domino plugs will actually fit into the socket two different ways round as the socket has a groove down each side and the plug has a corresponding lug which will go into either groove! You therefore need some way of remembering which way round you have wired the plug. Fortunately, most domino plugs has a gap in the metal shielding around the pins in such a way that when the plug fits into its socket, the gap is either uppermost or lowermost. So decide at the outset that you'll always plug in with the gap uppermost, as most people

seem to do, and wire accordingly. If your plug does not have a gap in the shielding, a couple of snips with a pair of side-cutters can soon create one!

Carefully join up RX's to TX's, CTS's to RTS's and earth to earth. Next join up pins 1, 4 and 6 of the D-type socket. If you're using bare wire to strap across these pins, make sure that it can never accidentally make contact with any of the other pins.

The additional DIN plug and socket are not connected together, but will be used to make a 'self-test' kit. In each case, link CTS back to RTS and TX back to RX in each, again making sure that neither link touches the earth tag.

Finally complete the job using a continuity tester to ensure that all connections are sound and that no neighbouring pins are touching each other.

Bauds and Bytes

Information flows through the RS423 ports in byte units. However, there is a choice of 8 different byte formats and 8 different transmission speeds. Although it's not necessary for our present purpose to understand these various options, one condition is essential: both machines must agree on the byte format and transmission speeds they are using. Since these are not necessarily identical 'ab initio' we'll want to know how to change them.

[Aside for Master users only: As a preliminary assessment, you can key in the following on both machines: *STATUS DATA and the machine then replies with an integer, say d, then do *STATUS BAUD and it returns, say b.

If both computers return the same value for d, and also agree on b – then their communication states are compatible. Otherwise you'll have to alter one or other of them. You can do this by

using the *CONFIGURE facility. But for simplicity, our program examples will make no assumptions about initial baud and data states, but will rather make them identical using *FX calls. This will ensure that any configuration states of the machines are not interfered with.]

***FX7,b** is the command used to set the baud rate for receiving data, while ***FX8,b** does the same for transmitting. 'b' is an integer 1 – 8: you can look up the corresponding baud speeds on p.389 of the Archimedes User Guide. One word of warning. The Archimedes has only one baud rate generator used for both receiving and transmitting. It is therefore recommended that you use the same value of 'b' in both *FX7 and *FX8. The following examples happen to use b=4 (1200 baud), but the user should feel free to experiment with other values.

***FX156,d,0** sets the RS423 communication state. You won't find it referred to in either User Guide (apart from the table on p.382 of the Archimedes, and page D.2-47 of the Master Reference Manual). There are some details buried deep in the massive Archimedes Programmers' Reference Book. You'll have to take it on trust that a value of d=&56 (hexadecimal 56) will set the data format to the commonly employed 8n1 (that means 8 bits/byte, no parity bit, 1 stop bit). Users who can decipher the obscurities are very welcome to try other values of d, at their own risk!

The other two *FX calls we'll be using are probably more familiar. ***FX2,n** specifies the input stream. We shall use n=0 for keyboard, and n=1 for RS423 (see Archimedes UG p.387). ***FX3,n** specifies the output stream. Of the eight options we'll be using only, n=0 output to screen and not the RS423, n=5 output to screen and RS423 and n=7 output to RS423 only

Testing, Testing!

The program listing on the next page is a BASIC program designed to test both the separate ports and your hardware connections. The identical program should be keyed into both machines.

If you leave both RS423 connectors unattached and RUN the program on either machine, the result appears to be a frozen keyboard. What's going on? When a key is depressed, the machine accepts the key character as input, but instead of sending it to the VDU screen, it routes output to the RS423 port. Since the latter is unattached, your key depressions are lost to space.

Next we use our self-test kit. Insert the DIN self test plug into BBC Micro's RS423 DIN socket. If you now key in some characters they should be echoed on its VDU display: true, by a circuitous route – out to the RS423, back again through the self-test connector, and then on to the screen! Now take the D-type to DIN connecting lead you've wired up, and insert the D-socket into Archimedes' RS423 D-plug. Lastly insert the DIN self-test socket onto the DIN plug at the other end of the lead. Keying in characters on the Archimedes keyboard should cause them to appear on the screen.

Cross Test

First remove both self-test DIN connectors, and insert the DIN plug at the end of your connecting lead into the BBC Micro's RS423 socket. Get the self-test program running on both machines and, all being well, anything keyed on the BBC Micro console should appear on the Archimedes screen and vice versa. This includes normal keys and also control codes such as <ctrl-L>, <ctrl-J> etc. So that keying in <ctrl-G> on either console will cause the other machine to bleep.

However, the ESCAPE key has to be specially recognised, in order to permit an orderly exit from either program. This is done by using an error handling routine.

```

10 ON ERROR PROCerror:END
20 MODE7
30 *FX7,4
40 *FX8,4
50 *FX156,&56,0
60 :
70 REPEAT
80   PROCin
90   PROCo
100 UNTIL0
110 :
120 DEFPROCo
130 *FX2,2
140 *FX3,7
150 A% = INKEY(10)
160 IF A% > 0 PRINT CHR$(A%);
170 ENDPROC
180 :
190 DEFPROCin
200 *FX2,1
210 *FX3,0
220 B% = INKEY(10)
230 IF B% > 0 PRINT CHR$(B%);
240 ENDPROC
250 :
260 DEFPROCerror
270 *FX2,0
280 *FX3,0
290 REPORT
300 PRINT " at line "; ERL
310 ENDPROC

```

Program Notes

- 20 Both machines set to same VDU mode to ensure compatible display of certain mode-dependent characters (e.g. `)
- 30,40 selects 1200 baud as the data speed
- 50 Data type 8n1: other values to be tried with great caution
- 70-100 Repeat input and output routines ad infinitum
- 130 accept input from keyboard but keep RS423 input buffer enabled
- 140 Send output to RS423 port rather than to VDU
- 150 Wait 0.1 sec for a key depression
- 160 If there is a character, transmit it. The semi-colon is required to ensure that only the bare character, without any spacing or line feed bytes is transmitted.
- 200 Switch to accepting input from RS423 instead of keyboard
- 210 and send output to VDU
- 220 Wait for 1/10 second for a character to appear on RS423
- 230 if there is one, send it to screen
- 270 If an escape character is detected: restore input from keyboard
- 280 and output to VDU
- 290 report, just in case there was an error in the program

Program transfer

Having got the communication system working, it is now time to get it to do some really useful jobs. Let us assume we have a program in the BBC Micro's memory and want to transfer it to the Archimedes. First put the Archimedes into a suitably receptive mood:

```

*FX7,4
*FX8,4
*FX156,&56,0
*FX2,1

```

(The first three commands are only necessary if you have not yet run the test program.) Then on the BBC Micro, you key in:

```
*FX7, 4
*FX8, 4
*FX156, &56, 0
*FX3, 7
LIST
```

(Again, the first three commands may not be necessary.) The program text will then miraculously appear on the Archimedes' screen! Moreover, it is also patiently sitting in Archimedes' RAM, waiting to be RUN or whatever. Unfortunately you can't get at it yet since the Archimedes' keyboard is frozen out. Do a <ctrl-break>, then key in OLD<return> and you're back in business. The program can be listed, edited, run, saved to disk etc.

Similar techniques will allow you to transfer any block of text (e.g. as generated by a word processor) and of course the same trick works in reverse – transferring from the Archimedes to BBC Micro.

Disk to Disk

The above technique could also be employed to transfer disk files, but it is much easier to do it by using a BASIC program and there is the added advantage of being able to copy any files whether they are text files or not.

The transmitting program first finds the length of the file in bytes, NB%, and transmits the number NB% (split up into a 4 byte integer) to the receiver, the latter is programmed to expect the first 4 bytes received to be a file length integer. Thereafter it counts bytes received and closes the file when the count reaches NB%.

Both programs maintain a 1 byte checksum, CHK%, of bytes transferred. After transmission

is complete these are printed out. If they agree, the two files are identical to a high degree of probability.

```
10 REM FILE TRANSM'N PROGRAM
20 CLS
30 ON ERROR PROCerror:END
40 INPUT"Source filename? "F$
50 X$=OPENIN(F$)
60 NB% =EXT#X%
70 PRINT"This file contains
";NB%;" bytes"
80 *FX3, 7
90 *FX7, 4
100 *FX8, 4
110 *FX156, &56, 0
120 N% =NB%
130 FOR I% =1 TO 4
140   VDU N%:N% =N% DIV 256
150 NEXT
160 CHK% =&AA:I% =0
170 REPEAT
180   I% =I% +1
190   Z% =BGET #X%
200   CHK% =CHK% EOR Z%
210   PRINT CHR$Z%;
220 UNTIL EOF#X%
230 *FX3, 0
240 CLOSE #X%
250 PRINT"EOF after ";I%;" bytes transferred"
260 PRINT"Checksum = ";CHK%
270 END
280 :
290 DEF PROCerror
300 *FX3, 0
310 CLOSE#0
320 REPORT
330 PRINT" at line ";ERL
340 ENDPROC
```

File transmission program notes

40 Ask user for name of source file
 50 Open it for reading (if no such file exists, an error will be generated)
 60 Get length of the file in bytes
 70 and print it out on the screen
 80-110 Set output to RS423, 1200 baud, data type 8n1
 120-150 Split length NB% into 4 bytes and send one at a time, starting with least significant byte
 160 Initialise byte count and checksum
 180 count number of bytes
 190 get next byte from source file
 200 update checksum
 210 send byte over RS423
 220 keep going until end of source file
 230 restore output stream to VDU
 240 close source file
 250-260 print messages

```

10 REM FILE RECEPTION PROGRAM
20 CLS
30 ON ERROR PROCerror:END
40 INPUT"Destination file
name? "F$"
50 X%=OPENOUT(F$)
60 *FX7,4
70 *FX8,4
80 *FX156,&56,0
90 *FX2,1
100 B0%=GET : B1%=GET
110 B2%=GET : B3%=GET
120 NB%=B0%+256*(B1%+256*(B2%+
256*B3%))
130 PRINT"Number of bytes = "
, NB%
140 PRINT"Received"
150 CHK%=&AA
160 FOR I%=1 TO NB%
170   Z%=GET
  
```

```

180   CHK%=CHK% EOR Z%
190   BPUT #X%, Z%
200   PRINT TAB(20,4),I%
210   NEXT
220   *FX2,0
230   PRINT"File closed with ";
EXT#X%;" bytes"
240   PRINT"Checksum = ";CHK%
250   CLOSE#0
260   END
270   :
280   DEF PROCerror
290   *FX2,0
300   CLOSE#0
310   *FX15
320   REPORT
330   PRINT" at line ";ERL
340   ENDPROC
  
```

File reception program notes

40 Ask user for name of destination file
 50 Open the file for writing
 60-80 set 1200 baud, data type 8n1, input from RS423
 100-110 get initial 4 bytes
 120-130 Reconstruct file length NB%, and print it out
 150 Initialise checksum
 160 for NB% bytes,
 170 get next byte from RS423
 180 update checksum
 190 write byte to destination file
 200 print running count of bytes received
 220 restore input from keyboard
 230 print file size to check same as NB%
 240 print checksum
 250 close destination file
 290-330 if an error occurs (or escape is pressed), restore input to keyboard, close all files, empty all buffers and print the error message.

Analog and User Ports

For many BBC users the existence of the analog and user ports is one of the most attractive features of the Acorn design – setting it uniquely apart from all competitors. They give the user control over a host of external devices – plotters, digital pads, measuring equipment, musical instruments, amateur radio signalling, RTTY decoding, stepping motors – the list is endless! Sadly these ports have been demoted to the status of optional extras on the Archimedes.

And now for the good news! If you have a BBC or Master you can easily run their ports from the Archimedes. Readers who have followed the arguments so far should have little difficulty in writing the necessary routines. These would involve the exchange of a series of bytes under a protocol in which the first byte was understood to be control information, and the rest data. The value of the control byte would determine which port was addressed, and whether it was to be written (e.g. ?&FE60= data) or read (e.g. ADVAL(1)).

However there is an alternative approach. Instead of individual bytes, the PRINT instruction can transfer an entire string over the RS423 – moreover this string could well be in the form of a valid instruction to the receiver's operating system, a BASIC instruction, or even an entire BASIC program.

Consider the following situation, in which we assume that both machines are already in baud and data compatible modes. You type, on the BBC Micro: *FX2,1 and on Archimedes:

```
*FX3,7
PRINT "10 A=0"
PRINT "20 B=1"
PRINT "LIST"
PRINT "RUN"
```

The effect would be to set up a simple two line program on the BBC, LIST and then RUN it.

The next step is to get the BBC Micro program to send any results of its running back to Archimedes. For this we shall need an extra *FX3,7 in the BBC Micro, and on the Archimedes carefully placed *FX2,1 and INPUT instructions. We have got a program working by this method which reads the ADVAL(1) on the BBC across to the Archimedes, but it is a bit tortuous and so we are working on a full implementation which will allow you to read or write to any of the I/O space on the BBC and also read the ADVAL values. We will hopefully be publishing the results next month.

[P.S. for Master owners. – Having opened up these exciting vistas, it would be irresponsible to fail to warn that all is not sweetness and light! There are some thorns among the roses, the sharpest of which lies in the state of the weather! Some Acorn Masters RS423 hardware appears to overheat, and malfunction in hot weather. Owners may have noticed this when in communications mode to Prestel or one of the Bulletin Boards, and attributed it to line noise. If the problem only develops after the computer has been switched on for some time, and only in warm weather, then you (like me) probably have an overheating problem. This will be confirmed if the malaise is cured by operating with the lid off, and fanning vigorously. I must admit I have not got to the bottom of this matter, since a finger-tip test reveals no undue heat in any chip. Comments from readers with similar experience, or who can throw any light would be very welcome.

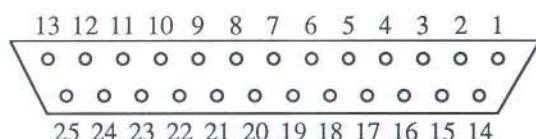
Even here there is a solution. It appears that only the top nibble of some bytes flowing over the RS423 get corrupted – whilst the bottom nibble

remains invariably intact. If readers' experience is the same then there's an obvious answer. Send each byte as 2 consecutive bytes, whose bottom nibbles are the top and bottom nibbles of the original byte: then reconstruct the original byte at the receiving end. It sounds rather silly, but it works!] 

Next Month: The BBC micro as an I/O podule to the Archimedes!

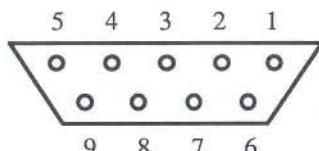
Pin connections for Printer Interface and RGB Output

(Each viewed from outside the back of the case, looking in.)



Parallel Printer Interface

1	STE		
2	D0	14	N.C.
3	D1	15	N.C.
4	D2	16	N.C.
5	D3	17	0V
6	D4	18	0V
7	D5	19	0V
8	D6	20	0V
9	D7	21	0V
10	ACK	22	0V
11	BSY	23	0V
12	N.C.	24	0V
13	N.C.	25	0V



RGB Output

1	RED		
2	GREEN		
3	BLUE		
4	CSYNC		
5	N.C.		
6	0V		
7	0V		
8	0V		
9	0V		

IEEE488 INTERFACE PODULES FOR THE ACORN ARCHIMEDES AND SPRINGBOARD COMPUTER SYSTEMS

The combination of Acorn 32-bit RISC Technology and Intelligent Interfaces hardware and software design results in an extremely fast and powerful IEEE488 Interface Controller for ATE (Automatic Test Equipment) systems.

- Full implementation of IEEE Std 488.
- Enables the computer to communicate with up to 14 IEEE488 interface compatible instruments
- Extremely fast data transfer rate (around 250 Kbyte/s)
- Can be configured as the system controller or as a simple talker/listener
- As controller in charge can send all addressed and universal commands and conduct both serial and parallel polls
- Ultra high level software interface for ease of use BBC BASIC Procedures and Functions supplied on disc optional libraries of procedures for FORTRAN 77, ISO Pascal, C etc
- Comprehensive user guide

Intelligent Interfaces supply IEEE488 Interface Controllers based on the Acorn Archimedes or the Compaq Deskpro Computer Systems. They designed, developed, manufacture and support the Acorn IEEE488 Interface for the BBC Models B, B+ and Master 128 Microcomputers.

Officially appointed Acorn Scientific Dealer.



Intelligent Interfaces Ltd

43b Wood Street

Stratford-upon-Avon

Warwickshire

CV37 6JQ

Tel: 0703 261514 or

0789 415875 (Head Office)

Telex: 312242 MIDTLX G



ROM-LINK on the Archimedes

James Lynn, Computer Concepts

One of the most common questions asked about Acorn's new Archimedes computer is "Will it run my old BBC software". Well, the definitive answer to this question is "Maybe!". If a program is written entirely in BBC BASIC, does not call any external machine code routines and does not access memory directly (for instance to save screen memory), then you can load it straight into BASIC V on the Archimedes and it will run straight away, and much faster than it ever did on the BBC.

If your program contains assembly language, however, BASIC V will not run it, as its assembler can only assemble the new ARM instructions, not the old 6502 instructions. To run a program of this type, Acorn supply a program called a 6502 emulator on the welcome disc. This is a program written in ARM assembly language that acts like a 6502 interpreter, so you can run any 6502 machine code programs as if they were running on a 6502 second processor. This includes the old BASIC IV which is the language in which the emulator starts up. The emulator can run any properly written 6502 machine code program. However much commercial software will not run in the emulator, since they tend to do illegal things such as accessing screen memory directly for speed. Such software must be rewritten before it will run in the emulator.

It is into this last category that the ROM-LINK series falls. Inter-Sheet, Inter-Word and Inter-Chart all had to be rewritten to a greater or lesser extent, before they would run on the Archimedes under the emulator. Inter-Base cannot be transferred because it comprises a total of 64K of code, and this would fill up the total memory

available under the emulator. Incidentally, this points up the major disadvantage of the emulator:- Since it acts like a 6502 second processor, you cannot make use of the whole 512k (or 1M) memory of the Archimedes. The 6502's fundamental limitation of 64k still exists. This article details the differences between the ROM-LINK series on the BBC and on the Archimedes.

The first one to be converted was Inter-Chart. This is the most 'legal' of the ROM-LINK series, as it only uses the standard operating system drawing commands, and so it has undergone the fewest changes. The ROM-LINK parts have had to be removed, due to the fact that you cannot run service ROM software in a second processor environment, and ROM-LINK relies heavily on the combination of service ROM and language ROM. The only extensions are the capability to use most of the extra graphics modes on the Archimedes, and the extra memory gained by relocating the program higher than it would be on a normal BBC.

Next program to be converted was Inter-Sheet. While not being quite as legal as Inter-Chart, it was already able to use legal screen update calls. As with Inter-Chart, the ROM-LINK parts of the package have been removed and the program has been relocated to give some extra memory. The other enhancement has been in the variety of screen display. On the BBC, we were able to offer 105 columns by writing thinner characters directly to the screen. This option was not really possible on the Archimedes, since a program running in the emulator cannot access screen memory at all, but instead, the new 132 column mode has been used. A fourth mode has also been added, using MODE 20. This gives 80 characters by 64 rows, and allows more than twice the depth of spreadsheet to be displayed.

However, most people will not be able to use this mode, as it needs a very expensive multi-sync monitor, which can cost £600 or more. For the lucky few who have such monitors, the 64 row mode is a useful addition.

Next to be converted was Wordwise Plus. In many ways, this is the least satisfactory conversion because Wordwise Plus works almost entirely in mode 7. There are several reasons for this. Firstly, when the original Wordwise was written, soon after the BBC micro was released, most people were still using ordinary televisions for their display, so a word processor that worked in an eighty column mode would not be very legible. Secondly, because screen modes 0 to 6 are bit-mapped modes they take a lot longer to update than a mode 7 screen, which is character-mapped. Thirdly, mode 7 only takes up 1K of the BBC's 32K memory. The next smallest screen mode, mode 6 takes up 8K. Thus, when using Wordwise, you always get a legible screen display, a fast screen update speed and the maximum amount of free memory.

However, on the Archimedes, these considerations are all virtually reversed. You cannot attach an ordinary television to an Archimedes, so we can assume 80 or even 132 columns will be readable. Also, mode 7 on the Archimedes is a bit of a cheat, as it has none of the teletext character generating circuitry, so it uses mode 9 to emulate all of the features of a teletext mode. This means that (a) it takes up more memory than any mode on the old BBC ever did, and (b) its update speed is as slow as any other mode. Thus, all three reasons for using mode 7 in Wordwise are negated. However, this was not seen as necessary to alter Wordwise Plus drastically, so it still uses mode 7 for editing. The main difference most people will notice is that the screen update is a lot slower than on the BBC.

However, it is offset by the fact that you can type more text than you ever could on a BBC since the Archimedes version is more like a Hi-Wordwise Plus, making use of the spare memory you get in a second processor. Also, any Wordwise Plus segment programs should run perfectly in this version, although the Archimedes has no ROM sockets as such, so only disc-based software will be usable.

On now to Inter-Word, the final program to be converted. This is the most obviously different conversion, and was the one which took most work. The main stumbling block was that Inter-Word on the BBC achieved its fast screen update in 80 columns by writing directly to screen memory. Since this is impossible to do from the emulator, and the legal Acorn screen drivers are a little too slow, some way had to be found to speed up Inter-Word's screen handling.

The method finally adopted was to use a relocatable module. This is a program, written in ARM machine code, which acts in a similar way to the way sideways ROMs act on the BBC. This relocatable module is called by Inter-Word while it is running, and is able to take over all the screen handling in Inter-Word's edit mode.

When originally written, it used mode 0 for the eighty column mode and mode 4 for the forty column mode, as on the BBC version. But then we thought it would be a good idea to use mode 16 to provide 132 columns instead of the old 106 columns on the BBC. This almost proved a problem, because mode 16 is a sixteen colour mode, and modes 0 and 4 are only two colour modes. This would mean two separate routines for the two types of mode, including two separate character sets. However, it was soon realised that the best way to avoid this overlap was to use modes 9 and 12, which are sixteen colour 40 and 80 column modes respectively.

After it was decided to use sixteen colours for the screen display, the logical progression was to make some use of the extra colours in the screen display. Thus, when starting up Inter-Word on the Archimedes, the user is presented with a cyan ruler, with the tab indicators in yellow and, in place of the cross-hatched area which so plagued many users on the BBC, the unused page area is now a solid grey colour. Also, bold text is a different colour from normal text, making bold text stand out a little more, and the underlining uses yet another colour. Just in case the pre-selected colours are not to the user's liking, a brand new menu has been supplied, allowing the user to change the pre-set colours to anything he or she wants, by adjusting the amount of red, green and blue in each colour. Thus, it is possible to have the screen displayed in delicate pastel shades, or in the most garish combinations imaginable, according to your own personal taste. The selected colour combinations are saved when you save your document to avoid having to re-set them.

Other enhancements include the use of the 80 column by 64 line mode (though this only works with a multi-sync monitor) allowing a whole page to be visible on screen at once.

All things considered, using the ROM-LINK series on the Archimedes offers no significant advantage over using them on the BBC, apart from some extra memory. However, since many users who upgrade from the BBC are familiar with them, and since it will be a while before the new generation of software appears on the Archimedes, Inter-Word, Inter-Sheet, Inter-Chart and Wordwise Plus are an ideal solution to the problem of what to do with a new machine before the specially written software becomes available. **A**

Archimedes Toolkit Module

Supplier: Clares Micro Supplies
98 Middlewich Road
Northwich
Cheshire
CW9 7DA
0606 48511

Cost: £39.95 (£37 to Archive members)

The Archimedes Toolkit Module is the first available toolkit for the Archimedes and the version reviewed here is version 3.50. This is a pre-release version costing £50 but by the time you read this, the production version should be available. I like the idea of providing a pre-release version because, although, on their own admission, there are "a few minor bugs" in it, those who really need this kind of toolkit for help in their own development work can get hold of it, albeit at £50 instead of the full release version price of £39.95. Those who buy the pre-release version will apparently receive a free upgrade as soon as it is ready.

The module is supplied on a 3.5 inch disc and comes with a plastic comb bound, daisy-wheel printed manual which was reasonably clear and readable. It divides the utilities up into two main groups, 'GENERAL' and 'ADFS'. You can get a list of the available commands by typing *HELP GENERAL or *HELP DISC. The general utilities include memory edit, search and compare and an option to send memory dumps to a printer in any of five formats (binary, hex, text, mnemonic or 32 bit word). The second section, which deals with the ADFS editor and searches, also provides CATALL and EXALL. Unlike the BBC's operating system, Arthur doesn't automatically show the star commands available for a module but you can find out by typing *HELP .<return> which shows all the commands for all the modules. This is useful as

it gives the full syntax and some useful information.

General Commands

The SHIFT command moves a block of memory from one location to another and the manual explains how to move and save a module, though not having written many modules myself yet(!) I haven't tried it out. ENVIRONMENT gives general information about the machine such as end of application workspace, error buffer location and the addresses for such things as the error handler, event handler etc. This is one for those who really know their way about the machine.

Then there are the printer dumps in the five formats mentioned earlier and a set of search routines using hex, text, mnemonic or 32 bit word formats and these routines can include wildcards in the search string. The text search, TFIND, can contain single and multiple

wildcards and the search can be set to be case-sensitive or case-insensitive. All searches can be started from any address by including the address in the command. The hex search, HFIND, takes a hex list as its input and this can contain single wildcards.

The two searches which would be unfamiliar to BBC toolkit users are MFIND and WFIND which search for a mnemonic or a 32 bit word respectively. MFIND allows you to search for a line of assembler. So, for instance, if you wanted to find all occurrences of OS_Byte you could do it by entering:

*MFIND "SWI OS_Byte"

and it would search memory, stopping to display each occurrence. When you stop, the search is in the memory editor and all of the editor options (described below) are then available. As with the searches in Computer Concepts' Disc Doctor, having edited that occurrence you can choose to

```

==> Help on keyword GENERAL
Archimedes ToolKit Module v3.50 (August 19th 1987)

*BIT <start> [[+]<end>]
*ENVIRONMENT
*HCOMPARE <start> [+]<end> <with>
*HEX <start> [[+]<end>]
*HFIND <hex list> [<address>]
*MEMORY [<address>]
*MFIND <mnemonic> [<address>]
*MNEMONIC <start> [[+]<end>]
*SHIFT <start> [+]<end> <destination>
*TEXT <start> [[+]<end>]
*TFIND <string> [<address>] [C]
*WCOMPARE <start> [+]<end> <with>
*WFIND <word> [<address>]
*WORD <start> [[+]<end>]

```

go on to the next match by pressing <ctrl><tab>.

The WFINDD command is even more useful because if you know the opcode numbers, you can search for any instruction and using the wildcards you can mask out condition flags etc. For instance to find all SWI calls you enter:

```
*WFINDD #F#####
```

The top four bits of the word identifies any conditions set for the SWI and the 'F' is the actual value that identifies that it is a SWI call. If you only wanted to find the SWI's without any condition codes then you would search for EF##### instead. Once you identify the particular SWI numbers you can then search explicitly for them, e.g.

```
*WFINDD #F000006
```

will find all occurrences of SWIOS_Bye which is SWI 6.

The final command in this section is just called MEMORY and is the entry point to the memory editor and a single line assembler. The format for the editor is configurable, as are many other options for the toolkit – the configurations are held on disc and loaded with the toolkit. When editing memory, you can use any of the five formats described earlier, the default format being the familiar hex with ASCII to the right. Once in the editor environment you can switch formats by single control keystrokes, <ctrl-B> for bit mode, <ctrl-M> for mnemonic etc. So if you do a word search for SWI 6 as above you can then press <ctrl-M> and immediately see the SWI OS_Bye in the disassembler.

If you press any alphanumeric key, you automatically enter the single line assembler. This can be used to assemble almost any valid line of assembler code. There are a few limitations which are well documented. The line is checked on input and if it is invalid, the

original line is left untouched. I found this a bit frustrating because if you don't get the code exactly right, the display returns to the editor giving no indication of why your entry was unacceptable. You don't know whether you had just mis-typed it or whether you were using an illegal assembler mnemonic. The advantage of the single line assembler is, of course, that you can make an alteration to your code without having to re-assemble the source code and this can obviously reduce debugging time, but the disadvantage is that you have to make sure you don't forget to modify the assembler listing!

ADFS

The disc commands are in many ways similar to the memory commands including the editors and single line assembler. I don't know of any other software that allows you to look at your disc in all the five formats and I guess that at times it could be very informative. The editor is entered with *AEDIT but you can also specify a drive and sector offset where the editing should start. Apart from this, the editor and search facilities are the same as the memory counterparts.

CATALL and EXALL round off the disc section and provide a useful means of surveying the contents of your disc directories. All the disc commands work with 640k and 800k floppies as well as, apparently, the winchester disc, though I haven't yet been able to test the latter facility!

DIY utilities

If you use the *HELP . command or *HELP SWI, it gives you a list of SWI calls relating to the toolkit. These include Assemble, Shift, Compare, DumpLine etc and these calls should be usable by anybody who knows the entry and exit parameters. Clares will apparently be supplying this information to interested parties

in the form of a technical manual. Clares have also said that they are going to add some more commands to the final version including a sector load and save, so when we get the full release version, we will say something about that and any other extra facilities they decide to add. (I also hear rumours that the production version will have a free buffer module on the disc.)

Summary

The Archimedes Toolkit Module is really quite impressive. Everything works at an incredible speed (but doesn't everything on the Archimedes?!?) so even on a large machine you can search through memory quite quickly. It

provides a wide range of the tools for the assembly language programmer and even non-programmers may find some of the routines valuable when trying to recover lost data or even just for exploring the machine. It is a shame, in a way, that they didn't choose to use the WIMP environment so that you could look at and compare different areas of memory, but it is early days yet in terms of Archimedes software development. Is it worth the price of £39.95 (£37 to full members of Archive)? Well, if you need a toolkit, there isn't much choice at the moment. As and when another toolkit comes along we will compare the two but until then, it's this or nothing! **A**

```
*HELP DISC
==> Help on keyword DISC
Archimedes ToolKit Module v3.50 (August 19th 1987)

*AEDIT <drive> [<sector address>]
*AHFIND <hex list> <drive> [<sector address>]
*AMFIND <mnemonic> <drive> [<sector address>]
*ATFIND <string> <drive> [<sector address>] [C]
*AWFIND <word> <drive> [<sector address>]
*CATAFF <current drive> [<path>]
*EXALL <current drive> [<path>]

*HELP GENERAL
```

Archimedes... the next Music Computer!

Mike Beecher of Electro-Music Research, well known for his virtuoso computer music demonstrations at major exhibitions writes:

Imagine the possibility of getting your music printed on screen as you play it and the opportunity to communicate with your keyboard – or, in fact, any MIDI-equipped instruments – by means of simple command words typed on the micro and you have taken a quantum jump into the next generation of computer-controlled music!

You may not have considered the Archimedes as particularly important to music and musicians, but there are others who think differently. ElectroMusic Research Ltd have been consulting closely with Acorn's development team on the Archimedes this year and the result is a "Musicians" Archimedes equipped with MIDI interface and a huge set of music and MIDI "commands" that lets virtually anybody who has dabbled with a computer – home or business, using BASIC or machine code – explore computer-aided music-making with ease.

What makes the Archimedes remarkable in terms of music-control power are features like the following:

- BASIC can run fast enough for many music programs to be created.
- Machine code runs so fast that 'multi-tasking' of recording, notation and graphic generation and interactivity between micro and musician can be achieved.
- Because WIMP commands are incorporated in BASIC it allows simple creation of professional style "windows" under mouse control and these,

combined with digitised pictures, are able to provide multi-level information sources for describing and illustrating music commands.

- This is the first micro to have both BASIC and machine code (32-bit) implemented with a comprehensive command set for controlling up to 8 powerful internal voices as well as MIDI instruments.
- Internal voices and MIDI communication can be interchanged intelligently so that just one or two MIDI instruments combined with internal voices could make an extremely versatile orchestra, band or other type of music ensemble.
- High quality sound sampling can be achieved with a suitable hardware podule, including fast fourier synthesis of waveforms on screen. Similarly, voice or other audio analogue input could be used in addition to micro keys or MIDI instruments.
- In terms of music education, the Archimedes' processing speed and direct access to internal/ MIDI instruments offers innovative possibilities for interactive programs that will encourage self-learning in the school and home.
- Computer-generated graphics for video communications linked with music are possible using the high-resolution screen modes.
- The large memory base allows music to be stored in compact coded format for more than enough compositional time on recording systems.

Of course, the world of MIDI is still undiscovered territory for many people and EMR will be opening their existing MIDI-hotline to Archimedes users on 0702 335747 in conjunction with Acorn to give advice. EMR are planning an exciting range of products for the Archimedes based on our very successful "Miditrack Music System" that uses the speed

and control of the 32-bit architecture to the fullest extent.

All the things we couldn't implement on the BBC micro due to lack of space, or control efficiently via MIDI, now appear possible and we'll be creating a system up to studio standard that's so easy to use, you'll find (like Miditrack) that youngsters will be able to record and edit their music played on a cheap Casio MIDI keyboard, print it out and send it to the Archimedes' built-in orchestra as well.

On the other hand there is tremendous potential for serious music use in creating complex music control, in the field of digital sampling, MIDI recording and in direct scoring from music. We'll be implementing some new features that actually encourage the musician to explore his music via the micro and even add his own ideas to the program in an intelligent way.

The Archimedes can have its own I/O podule that will take our existing BBC MIDI interface as a substitute for, or partner to, Acorn's MIDI

interface – this even has its own MIDI interrupt line! We are also likely to increase our sync facilities to include SMPTE time code for studio link-up.

Our program of development now includes a strong interest in systems for the handicapped and we still hope to maintain our emphasis on computer music programs that encourage creative music making rather than micro expertise:

From the first generation of home computers dedicated to music with Yamaha's CX51 and IIM, there is no doubt that we have, in the Archimedes, a superior system in terms of processing speed, available memory and music commands. – In short, a true music computer for today's music technology!

We'll try to keep you up to date on all developments musical with the Archimedes in future issues.

EMR's address is EMR Ltd, 14 Mount Close, Wickford, Essex, SS11 8HG.

Letters Page

Apart from various letters enquiring about the RS423 connection and setting up a 5.25" disc drive, we haven't yet had any letters – well, this is only the first issue! So if you have any moans and groans or clever ideas or comments about the Archive magazine, get those old word-processors going!

Structured Programming with BBC BASIC V

Dr Nigel Wilson

BBC BASIC has evolved from version I, as supplied with early BBC micros, through to version V which has appeared with the Archimedes range. The improvements up to version IV are many, but mostly corrections of minor bugs. Version V is the first to contain major improvements and, in particular, it provides the WHILE, CASE and block IF .. THEN control structures needed for implementing well-structured program designs.

The earlier versions of BBC BASIC contained only the REPEAT .. UNTIL and FOR .. NEXT constructs for loop control. BASIC V includes a WHILE .. ENDWHILE construct which, unlike the REPEAT .. UNTIL loop, tests the condition before the loop is entered. The following code requests a reply of 'Y' or 'N' from the user and, if any other character is entered, issues an error message and asks the user to try again.

```
PRINT "Answer Y or N"
INPUT ANS$
WHILE ANS$<>"Y" AND ANS$<>"N"
  PRINT"You must answer Y or N"
  INPUT REPLY$
ENDWHILE
```

The equivalent code using REPEAT .. UNTIL would also need an IF .. THEN construct, use of flags ('integers' taking the values TRUE or FALSE) or the nefarious use of GOTO's. Note that BASIC V does not insist on the use of line numbers. Because of the protection it gives against code being entered without the necessary conditions being met (in the above example, that an unacceptable reply has been

entered) the WHILE loop is generally more useful than the REPEAT loop.

The CASE statement can often be used to avoid several IF .. THEN constructs. For example, a menu routine may be needed which calls PROCA, PROCB or PROCC according to the user's response. The following code provides a neat, readable and easily modifiable solution :

```
PRINT "MENU"
PRINT "JOB A"
PRINT "JOB B"
PRINT "JOB C"
PRINT "SELECT A, B OR C"
INPUT CHOICE$
CASE CHOICE$ OF
  WHEN "A", "a" : PROCA
  WHEN "B", "b" : PROCB
  WHEN "C", "c" : PROCC
  OTHERWISE PRINT "ILLEGAL
                           CHOICE"
ENDCASE
```

where PROCA, PROCB and PROCC are defined elsewhere. Here CHOICE\$ is the selector variable whose value is compared with the lists of values (or expressions) following the WHEN's. As soon as a match is found, the instructions following the next colon are performed. When these are completed, execution continues from whatever follows ENDCASE. The instructions following OTHERWISE are only executed if no match is found. Whilst the OTHERWISE clause is optional, it is good practice always to include it so that illegal values of the selector variable can be trapped. The selector may be of any type providing the values or expressions following the WHEN's are of compatible types.

Some restrictions must be observed when using the CASE construct :-

a) WHEN, OTHERWISE and ENDCASE must be the first non-space object on a line.

b) CASE .. OF must be the last thing on a line.

The IF .. THEN .. ELSE .. construct provided with the earlier versions of BASIC must fit in one BASIC line which is limited to about 240 characters. This causes a problem if several statements need to be included in the THEN and ELSE parts. The problem can be avoided by putting these statements in procedures, called PROCtrue and PROCfalse say, so that we then have IF .. THEN PROCtrue ELSE PROCfalse which fits neatly in one line. Whilst the use of procedures is to be encouraged, the above method may result in numerous miscellaneous procedures having to be written and further problems relating to passing values to and from the procedures.

Another, and more serious problem, with the original IF .. THEN construct is that it cannot be properly nested. For example, in a routine to validate a date, some code is needed to check that the day is not too large for the month. The code to check that the date is not 31 June might be written as the single line:

```
IF MONTH$ = "JUNE" THEN IF DAY =
  31 THEN PRINT "Illegal day for
  June" ELSE PRINT "Legal day for
  June"
```

Unfortunately, if MONTH\$ = "MAY", say, then irrespective of the value of day, the output will be "Legal day for June" because the BASIC interpreter associates the ELSE part with the first IF rather than just the second IF as intended.

BASIC V provides a 'block' IF..THEN..ELSE .. ENDIF construct which may spread over as many lines as needed and can be properly nested. The ELSE part is optional. The above example can now be coded:

```
IF MONTH$ = "JUNE" THEN
  IF DAY = 31 THEN
    PRINT "Illegal day for June"
  ELSE
    PRINT "Legal day for June"
  ENDIF
ENDIF
```

where the indentation shows corresponding IF's, ELSE's and ENDIF's. In my view, it is clearer if the ELSE is given the same indentation as the corresponding IF and ENDIF statements, as above, but the indentation produced by using LISTO7 in BASIC V puts the ELSE in line with the two PRINT statements.

The following rules must be observed :-

a) ELSE and ENDIF must be the first non-space objects on a line.

b) THEN must be the last thing on a line.

The one line IF .. THEN .. ELSE .. construct is retained in version V for compatibility with earlier versions.

Passing parameters to procedures

Versions I to IV of BBC BASIC allow values to be passed to, but not back from, procedures by using parameters. This method is referred to as 'call by value'. Consider PROCSWAP which is designed to interchange the values of its two parameters :-

```
A = 1 : B = 9
PRINT "Before procswap : A = ", A, "B = ", B
PROCSWAP(A, B)
PRINT "After procswap : A = ", A, "B = ", B
END

DEF PROCSWAP(X, Y)
PRINT "At start of procswap : X = ", X, "Y = ", Y
TEMPORARY = X
X = Y
Y = TEMPORARY
PRINT "At end of procswap : X = ", X, "Y = ", Y
ENDPROC
```

Here A and B are called the 'actual parameters' and X and Y the 'formal parameters'. The output from the above program is

```
Before procswap : A = 1 B = 9
At start of procswap : X = 1 Y = 9
At end of procswap : X = 9 Y = 1
After procswap : A = 1 B = 9
```

and so swapping the formal parameters has had no effect on the actual parameters. If the procedure was to be of any use, the values of A and B should have been swapped upon returning to the main program. In BASIC V this can be achieved by putting RETURN before each of the formal parameters in the procedure definition heading :-

```
DEF PROCSWAP (RETURN X, RETURN Y)
```

When the procedure is called, the actual parameters will be copied into the formal parameters, as was the case previously, but now before the procedure is exited, the new values of the formal parameters will be copied back into the actual parameters and so the program output is :-

```
Before procswap : A = 1 B = 9
At start of procswap : X = 1 Y = 9
At end of procswap : X = 9 Y = 1
After procswap : A = 9 B = 1
```

PROCSWAP can now be used to swap the values of any pairs of parameters from any points in the program. This method of passing parameters is called 'call by reference'. However, when you are calling by reference, it does not make sense to supply a literal (as distinct from a variable) or an expression as an actual parameter. For example, the following are all illegal:-

```
PROCSWAP(1, A)
PROCSWAP(1, 2)
PROCSWAP(A + B, A)
```

With the features described above, BBC BASIC has been brought very close to the programming language Pascal whilst maintaining the convenience of an interpreted rather than a compiled language. Pascal still wins on variety of data structures and execution speed, but the appearance of BASIC V will be welcomed by all those concerned with writing well-structured programs. A

WITH
PAYROLL

A LOW COST DISC BASED
**PROFESSIONAL
ACCOUNTING PROGRAM**

For the BBC Computer

"Micro-Trader"

INTEGRATED SALES PURCHASE AND NOMINAL LEDGERS

Designed for business use by a business man. "Micro-Trader" offers full Sales and Purchase Ledger facilities including SALES INVOICE and STATEMENT PRINTING with a capacity of up to 1,000 accounts and 5,000 transactions per month.

Nominal Ledger features quick and easy entry of everyday transactions. Up to 6 Bank Accounts, full V.A.T. facilities, Journal Posting. Full Reporting for individual accounts, Audit Trail, Trial Balance, Profit & Loss and Balance Sheet.

"MICRO-STOCK"

Stock Control program, fully integrated with "Micro-Trader". 4,000 Stock Items with user defined Codes. Invoice and Credit Notes and Cash Sales routine. Full Stock Held Reports.

£75.00 + V.A.T.



"EXTENDED PAYROLL"

Micro-Aid Full Feature Extended Payroll. For use stand alone or integrated with "Micro-Trader".

£49.95 + V.A.T.

"MICRO-MAILER"

A Mailmerge program, fully integrated with "Micro-Trader". With a user Database.

£30.00 + V.A.T.



PHONE TODAY FOR A FREE FACT SHEET

MEADOW COMPUTERS

11, LONDON STREET, WHITCHURCH, HAMPSHIRE, RG28 7LH

Telephone: Whitchurch (0256) 892008

Technical Notes on BASIC Version V

Dr Andrew Bangham

BBC BASIC has seen a number of improvements since the innovative version I was introduced with the BBC micro. A number of bugs and inefficiencies were corrected to produce version II, the version of BASIC fitted to most B and B+ machines. A few minor modifications followed with version III, the next major release being version IV on the Master series of computers. The major changes related to screen editing facilities; trailing spaces are always stripped from lines entered into the interpreter; leading spaces stripped when in LISTO N (N>0); LISTO indentations follow practices of Pascal etc. For example, statements between but not including REPEAT and UNTIL are indented. The LIST command can be modified by including reference and search conditions (LIST 10,1000 IF A%) in addition to just line numbers, further editing facilities are provided by the inclusion of an EDIT command.

Other changes in version IV included:

- a real time clock psuedo variable TIME\$ that for example, prints as Wed, 31 Dec 1900. 23:59:59
- a 'l' terminator for VDU statements that completes the trailing zeros in the VDU drivers, for example, VDU19,1,2;
- ON expression PROCa, PROCb, PROCc ELSE PROCd became possible, though a bug in the ON ... command remains, namely that another statement following the ELSE clause is always executed. For example, in 'ON expression PROCa, PROCb, PROCc ELSE PROCd :PROCe' the PROCe procedure is always executed.

- The random number, rewritten to avoid statistical errors, was changed in a way that makes RND(1) and RND(N) where N>1 inconsistent.
- general recursion for FOR loops became allowable.

General changes in version V

Versions I to IV of BBC BASIC run on the 6502 series of processors. A major improvement occurred when the language was adapted to the Acorn RISC Machine (hence ARM BASIC) found at the heart of the Archimedes computer. The extensions include a large number of statements and functions concerned with sound, graphics and mouse control and, of course, the BASIC assembler handles ARM mnemonics and instructions instead of 6502.

A few of the important new grammatical constructions that have been added to BASIC are worth detailing. These include three new block constructions that were glaring omissions from earlier versions; IF .. THEN .. ELSE .. ENDIF, WHILE .. ENDWHILE and CASE .. OF .. WHEN .. OTHERWISE .. ENDCASE.

Two new unary operators are provided; binary constants can be represented as %1010101 (which, in this case, produces the same value constant as hexadecimal &55, or decimal 85) and there is a floating point indirection ! which is used to insert or retrieve real numbers from memory, e.g. !A% = 1.345 analogous with \$A% = "HELLO" or !A% = &89ABCDEF.

There are useful new binary operators; << shift left by the number of bits to the right of the operator, (bits are shifted left and zeros fed into the rightmost empty bits). For example...

X% = &DD

PRINT ~ (X%); "shifted="; ~ (X% << 3)
yields DD shifted = 6E8
i.e. 11011101 shifted = 11011101000

```

IF <expression> THEN           THEN must be last thing on the line
  <any number of statements/lines including nested IFs>
  ELSE <any number of statements/lines including nested IFs>
    optional ELSE must be first on line
ENDIF

WHILE <expression>
  block
ENDWHILE

CASE <expression> OF
  WHEN <expression>:<any number of statements/lines>
    [any number of WHEN statements]
  OTHERWISE <statements>
ENDCASE

```

Figure 1 – New Block Structures in BASIC V.

Remember that the ARM is a 32 bit machine that will handle 4 bytes at a time. The `>>` operator performs an arithmetical shift right, the leftmost bit remains unchanged unlike the `>>>` operator that performs a logical shift right by the number of bits to the right of the operator, the leftmost bit being set to zero. Thus `&FFFFFF>>3` remains unchanged but `&FFFFFF>>>3` gives `&1FFFFFF`. However, if the top bit is already zero, both `>>` and `>>>` have the same effect. (N.B. the second example on page 165 of the User Guide is incorrect. The operand in each case should be 1 and not 2 since in each case the bit pattern is shifted by one place only, not two.) Remember to ensure that brackets are used to force the order of evaluation when using multiple expressions containing these operators.

Extending Languages

The time taken to write programs tends to be proportional to the number of statements coded, whatever the language. Thirty lines of debugged assembler will probably take about as long as

thirty statements of BASIC. So using powerful, high level statements makes authors more productive; even more so if he can use powerful extra commands in the form of his own or other people's subroutines. With these, the author can "stand on their shoulders" and put into practice more complex ideas than would otherwise be possible. More powerful commands can easily be created by generating subroutines – define the subroutine once and you then stand on its shoulders each time you invoke it by a single, simple call statement. For a long time, languages such as FORTRAN, Pascal and C have provided methods of building subroutine libraries on an on-going basis. However, although BBC BASIC made some moves in this direction by permitting procedures and functions, it was never really enough. This left room for others to provide the necessary BASIC patches. "Overlay BASIC" from Elsevier Biosoft (see Fact-File), for example, allows procedures and functions to be stored on disk from where they will be retrieved automatically by any program that calls them. In other words it provides a

transparent library facility that, incidentally, uses overlays to make effective use of the limited memory in earlier computers. Some of these difficulties have been removed in version V.

Passing variables by value and by reference

There are major improvements in the way procedures and functions can be used. They can be built into libraries and there are tools for array handling. Most of the new features bring the language further into line with languages like C. For example, arrays can now be passed by reference and can be declared locally (the space being scavenged from the heap of variables on returning from the procedure or function). Ordinarily when a procedure (or function) passes a parameter, a copy of that actual variable is placed on the BASIC stack (which builds down from top of memory). Thereafter, that variable name may be safely used within the subroutine because the stacked value is reinstated on returning from the calling program. In this case the variable is passed by value. Incidentally, ordinary LOCAL variables are also stacked for the duration of the procedure and then popped off on return, though local arrays are stored on the heap (which builds up from the program instead of down from HIMEM). Whilst this is an excellent way of isolating variables within a subroutine from calling routines, it is over-restrictive. Functions allow a little more flexibility as single values can be returned from a subroutine, but if passing one value is good, passing more is even better. BASIC V allows any number, for example:

```
actualA=10:actualB=20
PROCSwap(actualA, actualB)
PRINT actualA;" ";actualB
END
```

```
DEF PROCSwap (RETURN A, RETURN B)
IF A>B THEN SWAP A,B
ENDPROC
```

The RETURN command instructs the interpreter to copy the values of the actual parameters into the local variables (A, B) as it enters Swap (note that it uses the new SWAP command). The procedure can then alter them before, on returning, the interpreter copies the new parameter values (values of A,B) back into the actual parameters. By copying values at each stage, the interpreter can make type conversions, say, between integers and reals.

BASIC V can also pass arrays to and from subroutines by reference. Instead of copying entire arrays each time a subroutine is entered and left, it is possible to pass the array address. This was possible using earlier versions, but very cumbersome as the array had to be found and the address temporarily inserted directly into the subroutine formal array name. In BASIC V it is possible to pass the address (reference) of the array to a subroutine by simply using an empty array notation, for example, the address of array X(10,10) is passed as X(). The formal procedure (or function) parameter name is expressed in the same way, and as a result the subroutine can operate on the elements of an array without ever knowing its original name (it only uses the local formal name). The advantages of using this technique are illustrated by the low-pass filtering routine opposite. This also illustrates a function that finds the number and size of array dimensions. The combination means that general purpose subroutines can be produced that perform operations on arrays, e.g., sorting, finite fourier transforms, filtering etc. (Note that although line numbers are not necessary for the functioning of well structured programs, ARM BASIC still requires them!)

```

DIM data(1000)
time1=TIME
FOR i=1 TO 1000
    data(i)=RND(1)+SIN(i)
NEXT i
time2=TIME
error%=FNlowpass(data(), 0.8)
time3=TIME
IF error%>0 THEN
    IF error%<1 THEN PRINT"Error: array must have one dimension"
    IF error%<2 THEN PRINT"Error: 0<k<1, (smoothing inc. with k)"
ELSE
    PRINT "It took ";(time2-time1)/100;" sec to create data"
    PRINT "It took ";(time2-time1)/100;" sec to filter the data"
ENDIF
END

DEF FNlowpass(x(),k)
LOCAL y() :REM an unnecessary complication for the filter itself
        :REM but this illustrates how it can be used
LOCAL i,n,error%
IF (k<0) OR (k>1) THEN
    IF DIM(x())<>1 THEN
        error%=1
    ELSE
        n=DIM(x(),1)
        FOR i=1 TO n
            IF i>1 THEN y(i)=y(i-1)*k+x(i)*(1-k):ELSE:y(i)=x(i)
        NEXT i
        x()=y() :REM transcribe filtered array
        error%=0
    ENDIF
ELSE
    error%=2
ENDIF
=error%

```

Figure 2 – Low-pass filtering program. This program takes 0.56 seconds to load the array plus 0.54 seconds to run the filter compared with 5.5 + 15 seconds on BBC Master, 33 + 16 seconds in Microsoft Basic on the Mac, 22 + 13 seconds on MacPascal, and approx. 0.07 + 0.07 seconds on the SUN 3/52 (68020 with 68881 maths coprocessor).

Array operations

BASIC V provides a large set of array operations including, unusually, a matrix multiplication operation in addition to element by element multiplication. The general syntax is to refer to arrays using empty brackets, for example;

`A()=B() + C` element by element addition of C to B() into A()

`A()=B() * C()` element by element multiplication of B() and C() into A()

`A()=B().C()` matrix multiplication

Library Facilities

The library facilities are less impressive. True, they are similar to other compiled languages, but one of the subtle, under-used but enormously powerful virtues of BBC BASIC is way the EVAL command can be used in situations where the programmer cannot anticipate what a user wants to enter from the keyboard. EVAL can evaluate functions which are declared "on-the-fly", something that is pretty difficult for a compiled language. Take, for example, a case where one wants to plot graphs of mathematical functions. In BBC BASIC, the user simply enters the equation into a string variable (on an INPUT A\$ statement) and it can be evaluated using EVAL(A\$). Try doing that in Pascal!

Furthermore whole BASIC functions can be evaluated, for example, if the program includes the function DEF FNdecay(Initialval, Timeconstant) the user could, as an alternative to keying a simple expression, key 'FNdecay(10,2)' into A\$. This becomes even more powerful if the program can automatically access a disk library of functions, perhaps with contributions from the user himself. This allows a program to automatically 'hook-in' functions written by the customer. It is this that makes the

Overlay BASIC patch for 6502 based BBC micros particularly powerful. It is a pity that version V cannot do this for it would have been better to build on the strengths of the BBC BASIC interpreter rather than emulating what compiled languages have done for years.

However, to look on the positive side, BASIC V does allow libraries to be searched for unknown procedures which can then be loaded into the heap or installed above the stack (above HIMEM). LIBRARY loads an internal form of a BASIC program onto the heap and INSTALL loads it above the stack (this cannot be done when anything is on the stack, i.e. from within a procedure/function or within a loop). Unlike the Overlay BASIC patch, Version V cannot handle any references to line numbers within a library procedure; it will simply jump to the host program and try to execute the specified line there instead. As mentioned above, the new structuring commands render line numbers unnecessary for anything except for editing.

Error handling

There has been a significant improvement in the way errors are handled. In the past, the interpreter reset the BASIC stack when an error occurred, so if an error occurred within a procedure/function or loop it was impossible to recover properly. Version V allows errors to be local; e.g. ON ERROR LOCAL PRINT "Error", and a whole block of code can be isolated by saving the error status (LOCAL ERROR) before and restoring it afterwards (RESTORE ERROR). The latter command can be omitted if it would fall just before a return from a subroutine as the 'ENDPROC' or '=' will do it anyway. A



MICROMAN **Computers**

**PASTURE LANE FARM,
PASTURE LANE,
RAINFORD, ST. HELENS,
MERSEYSIDE, WA11 8PU**

**PLEASE NOTE NEW
SATURDAY OPENING TIMES
9-6 MONDAY TO FRIDAY
9-1 SATURDAY**

Telephone
074488 3667

Three Graphics Demonstration Programs

Here are three quite nice graphics demonstration programs for you to try out. The first was sent in by Nigel Stuart of Fairhurst Instruments Ltd and is a line-art program which uses a set of 30 lines mirrored in the X and Y directions in random multiple colours, moving forward in randomly selected directions which produce a fascinating result. No comments will be made about the listing as it does not contain any special techniques, just to note that the VDU23,11 at line 30 is not mistake – it is explained in the “Have you discovered... ?” section on page 7.

The second program is a “one-liner” written by someone called Paul Beverley. RUN the program and see what happens as you move the mouse around.

Let’s have a few more one-liners from all you budding Archimedists. I’m not offering any prizes for the best, but let’s see what you can do.

The third program which was based on one on the “Creative Graphics” demonstration disc for the BBC micro which was produced yonks ago, was written by Adrian Look, a student at the University of East Anglia. It draws a coloured spiral which rotates back and forth as you press the space bar and then watch what happens when you escape from the program! **A**

```
10 K%=20:
W%=1279:
H%=1023
20 MODE 12
30 VDU 23,11
40 X1%=100:
X2%=200
50 Y1%=90:
Y2%=290
60 MM1%=K%:
MM2%=-K%
70 M1%=-K%:
M2%=K%
80 DIM W%(30),X%(30),Y%(30),Z%(30)
90 MOVE X2%,Y2%
100 REPEAT
110   X1A%=W%(1):
X2A%=X%(1):
Y1A%=Y%(1):
Y2A%=Z%(1)
120   X1%=X1%+M1%:
IF X1%>W% OR X1%<1 THEN
      M1%=-M1%
130   X2%=X2%+MM1%:
IF X2%>W% OR X2%<1 THEN
      MM1%=-MM1%
140   Y1%=Y1%+M2%:
IF Y1%>H% OR Y1%<1 THEN
      M2%=-M2%
150   Y2%=Y2%+MM2%:
IF Y2%>H% OR Y2%<1 THEN
      MM2%=-MM2%
160   GCOL 0,0:
MOVE X1A%,Y1A%:
DRAW X2A%,Y2A%
170   GCOL 1,7:
MOVE X1%,Y1%:
DRAW X2%,Y2%
180   GCOL 0,0:
MOVE X1A%,H%-Y1A%:
DRAW X2A%,H%-Y2A%
```

```

190  GCOL 1,6:
      MOVE X1%,H%-Y1%:
      DRAW X2%,H%-Y2%
200  GCOL 0,0:
      MOVE W%-X1A%,H%-Y1A%:
      DRAW W%-X2A%,H%-Y2A%
210  GCOL 1,5:
      MOVE W%-X1%,H%-Y1%:
      DRAW W%-X2%,H%-Y2%
220  GCOL 0,0:
      MOVE W%-X1A%,Y1A%:
      DRAW W%-X2A%,Y2A%
230  GCOL 1,4:
      MOVE W%-X1%,Y1%:
      DRAW W%-X2%,Y2%
240  W%(30)=X1%:
      X%(30)=X2%:
      Y%(30)=Y1%:
      Z%(30)=Y2%
250  FOR X=1TO29
260    W%(X)=W%(X+1):
      X%(X)=X%(X+1):
      Y%(X)=Y%(X+1):
      Z%(X)=Z%(X+1)
270  NEXT
280  IF RND(20)=1 THEN PROCch
290  IF RND(10)=1 THEN PROCrnd
300  IF RND(100)=1 THEN PROCCol
310 UNTIL0
320 :
330  DEFPROCrnd
340  IF RND(20)=1 THEN M1%=-M1%
350  IF RND(20)=1 THEN M2%=-M2%
360  IF RND(20)=1 THEN MM2%=-MM2%
370  IF RND(20)=1 THEN MM1%=-MM1%
380  ENDPROC
390  :
400  DEFPROCcol
410  C1=RND(175)+80:
      C2=RND(175)+80:
      C3=RND(175)+80
420  COLOUR 7,C1,C2,C3
430  COLOUR 6,C1,C2,C3-50
440  COLOUR 5,C1,C2-50,C3
450  COLOUR 4,C1-50,C2,C3
460  ENDPROC
470  :
480  DEFPROCch
490  RAN%=RND(15)+5
500  M1%=SGNM1%*RAN%
510  M2%=SGNM2%*RAN%
520  MM1%=SGNMM1%*RAN%
530  MM2%=SGNMM2%*RAN%
540  ENDPROC
550  REM (C) 1987 NIGEL STUART

```

Figure 1 – Nigel Stuart's Line-Art Program

```

10 MODE15:
  REPEAT:
  GCOL RND(64):
  MOUSE X%,Y%,Z%:
  C%+=RND(21)-11+20*((C%>200)-(C%<-200)):
  OSCLI"FX19":
  CIRCLEFILL X%,Y%,C%:
  UNTIL0

```

Figure 2 – Paul's “one-liner” – type it in, RUN it and wiggle the mouse about. Now see if you can think up a “one-liner” that is better than that.

```
10 REM ****
20 REM *          Spiral      *
30 REM *  By Adrian Philip Look  *
40 REM *          04/9/1987      *
50 REM ****
60 :
70 ON ERROR PROCerror:
    END
80 MODE12
90 VDU29,750;512;5
100 :
110 PROCinitialise
120 PROCdraw
130 PROCtwirl
140 END
150 :
160 DEFPROCinitialise
170 DIM value(666)
180 offset%=0
190 colour%=1
200 limit%=6
210 direction%=1
220 FOR angle%=0 TO 666
230 value(angle%)=SIN(RAD(angle%))
240 NEXT
250 ENDPROC
260 :
270 DEFPROCdraw
280 FOR radius%=512 TO 72 STEP -8
290     MOVEvalue(360+offset%)*radius%,value(450+offset%)*radius%
300     FOR angle%=8+offset% TO 360+offset% STEP 8
310         GCOL0,colour%
320         colour%=colour%MOD5+1
330         MOVEvalue(angle%+4)*(radius%-72),value(angle%+94)
                                         *(radius%-72)
340         PLOT&55,value(angle%)*radius%,value(angle%+90)*radius%
350     NEXT
360     offset%+=1
370 NEXT
380 ENDPROC
390 :
```

```
400 DEFPROCtwirl
410 start%=limit%-direction%
420 null$=CHR$(0)+CHR$(0)+CHR$(0)
430 REPEAT
440     change$=""
450     FORphysical=1TO5
460         logical=start%+physical
470         IFlogical>5THENlogical=logical-5
480         change$=change$+CHR$(19)+CHR$(physical)+CHR$(logical)
                                         +null$
490     NEXT
500     IFZ%=667VDU23,7,1,1|
510     *FX19
520     PRINTchange$|
530     start%=start%+direction%
540     IF start%=limit% THEN start%-=direction%*5
550     IF INKEY(-99) THEN PROCreverse
560 UNTIL 0
570 ENDPROC
580 :
590 DEFPROCreverse
600 limit%=limit%-direction%*6
610 direction%=-direction%
620 REPEAT
630 UNTIL NOT INKEY(-99)
640 ENDPROC
650 :
660 DEFPROCerror
670 IF ERR=17 THEN
680     IF Z%<>667 THEN
690         Z%=667:
                     PROCtwirl
700     ELSE
710         Z%=0
720         VDU4
730     ENDIF
740     ELSE
750     CLS
760     REPORT
770     PRINT" at line "; ERL
780 ENDIF
790 ENDPROC A
```

Mode Selection Chart

		Number of Colours			
Text	Graphics	2	4	16	256
20 x 32	160 x 256		5	2	10
40 x 25	(Text Only)	6		7	
40 x 32	320 x 256	4	1	9	13
80 x 25	(Text Only)	3	11	14	
80 x 32	640 x 256	0	8	12	15
132 x 25	(Text Only)			17	
132 x 32	(Text Only)			16	
80 x 64	640 x 512	18	19	20	

User Guide Errata

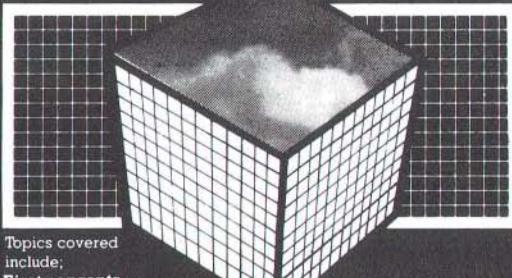
Let us know if you find any mistakes in the User Guide. We can then keep everybody informed.

- p33 It looks as if a whole section is missing as the program example stops in the middle.
- p87 line1 “It does not give no information” – oops!
- p165 middle All the “>>2” and “>>>2” should be “>>1” and “>>>1”
- p405 last line “shift-f2 (UNDO)” should be “shift-f10 (UNDO)”
- p408 The final statement is incorrect: <ctrl-R> **cannot** be used to remove line markers (well not on my version of ARMBE, anyway.)
- p456ff There is no index! Surely the manual for a £1,000 computer should have an index, shouldn’t it? Anyone fancy writing one?

ARM Assembly language programming

By Peter Cockerell

This book describes in detail all aspects of programming the new Acorn Risc chip (generally referred to as the ARM). This 32-bit chip, on average 4 times faster than a 68000, will form the basis of a new generation of machines. The book explains why the chip is so fast and how to take full advantage of its power.



Topics covered
include:

- First concepts.
- Inside the ARM.
- The ALU and barrel shifter.
- The instruction set.
- Instruction timings.
- The BBC BASIC assembler.
- Assembly programming principles.
- Data structures.
- Non-user modes.
- Undefined instructions.
- Interrupts and vectors.

The book is completely up-to-date covering only the new 2 micron ARM, with appendices on the floating point and co-processor instructions and with numerous example programs. A truly comprehensive reference manual for anyone who intends to program in ARM assembly language, or for those who just want to find out why the Acorn RISC chip is so special.

Price £12.95



Computer Concepts

Gaddesden Place, Hemel Hempstead, Herts HP2 6EX. Tel: (0442) 63933

Access/Barclaycard accepted. All orders despatched within 24 hours.

AVAILABLE
NOW!

Tail-Piece

Would anyone like to sort out any of the following and let me know so that we can publish the information for others?

1. Compare and contrast the effects of <break>, <ctrl-break>, <reset> and <ctrl-reset>.
2. What exactly is the function of the FILESWITCH module?
3. What exactly is the function of the SYSTEMDEVS module?
4. How exactly does the Debugger module work?

It's not supposed to be good to admit ignorance, but why not? It's just that I don't want to spend ages sorting these things out if someone else has already done so. That's what this magazine is supposed to be all about, isn't it?

Fact-File

Audio-Visual Productions Hocker Hill House, Chepstow, Gwent, NP6 5ER. (02912 - 5439)
CJE Micros Dept AM, 78 Brighton Road, Worthing, W Sussex, BN11 2EN.
(0903 - 213361)

Clares Micro Supplies 98 Middlewich Road, Rudheath, Northwich, Cheshire, CW9 7DA.
Computer Concepts Ltd Gaddesdon Place, Hemel Hempstead, Herts, HP2 6EX. (0442 - 63933)
Elsevier-Biosoft 68 Hills Road, Cambridge, CB2 1LA. (0223 68622)
EMR Ltd 14 Mount Close, Wickford, Essex, SS11 8HG. (0702 - 335747)
ESM Software 32 Bridge Street, Cambridge, CB2 1UJ. (0223 - 65445)
Intelligent Interfaces Ltd 14 Julius Close, Chandlers Ford, Eastleigh, Hants, SO5 2AB.
(04125 61514)

Meadow Computers Ltd 11, London Street, Whitchurch, Hants, RG28 7LH. (025689 - 2008)
Minerva Systems 69 Sidwell Street, Exeter, EX4 6PH. (0392 - 37756)
Tubelink PO Box 641, London, NW9 8TF.

Order Form

		Quantity	Total
File Transfer Service:	Send your 5.25" discs, DFS or ADFS and £5 per disc, inclusive, and we will transfer the files onto an ADFS 3.5" disc. (One 3.5" for each 5.25".)		
Program Disc	Vol 1.1 - £2 / Vol 1.2 - £3 / Vol 1.3 - £3		
Blank Keystrips	5 blank keystrips on A4 card - £1 per card		
Unformatted 3.5" Discs	Bulk pack, Wabash - 10 for £16 Wabash in library cases - 10 for £18		
Wordwise Plus - Full Package	£30		
Wordwise Plus - Upgrade	£10		
	(Please quote registration number.)		
CP-ROM on Disc - Full Package	£18		
CP-ROM on Disc - Up-Grade	£6 (Please return old CP-ROM)		
"ARM Assembly Language Programming"	£12		
Minerva Deltabase	£26		
Minerva System Delta-Plus	£64		
Minerva Minotaur	£13		
Clares' Archimedes Toolkit Module	- £37		
Clares' Imagewriter	- £27		
Clares' Artisan Art Package	- £37		
Programmers Reference Manual	£28		
Shareware Graphics Demonstration Disc	£3		

I enclose a cheque payable to "Norwich Computer Services" for:
(Sorry on Acces or Visa facilities.)

All prices are inclusive of VAT and UK carriage.

Name _____

Address _____



Norwich Computer Services, 18 Mile End Road, Norwich, NR4 7QY.
Subscription - £10 per year, Technical Help Service - £8 per year